**CS-470 Report**

**Group Members:**

Truman Delles

Christian Rodas

Ashish Sharma

Henri Ho

Spencer Apel

[OTHER TEAM MEMBERS FULL NAMES]

**Product Description**

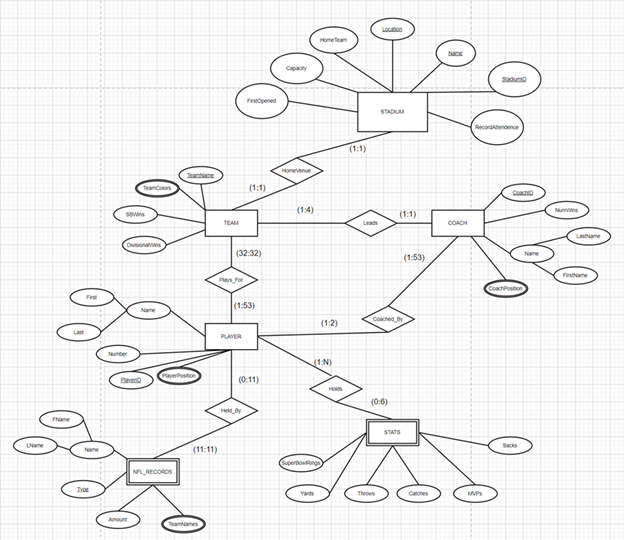
For our SQL Database project, we decided to go with a database of NFL Players, NFL Teams, NFL Stadiums, and NFL Coaches. Along with all necessary statistics and information that would be useful for each. The database has queries that can pull relevant information such as how coaches a player or which player has the most sacks so far.

**Requirements Summary of Project**

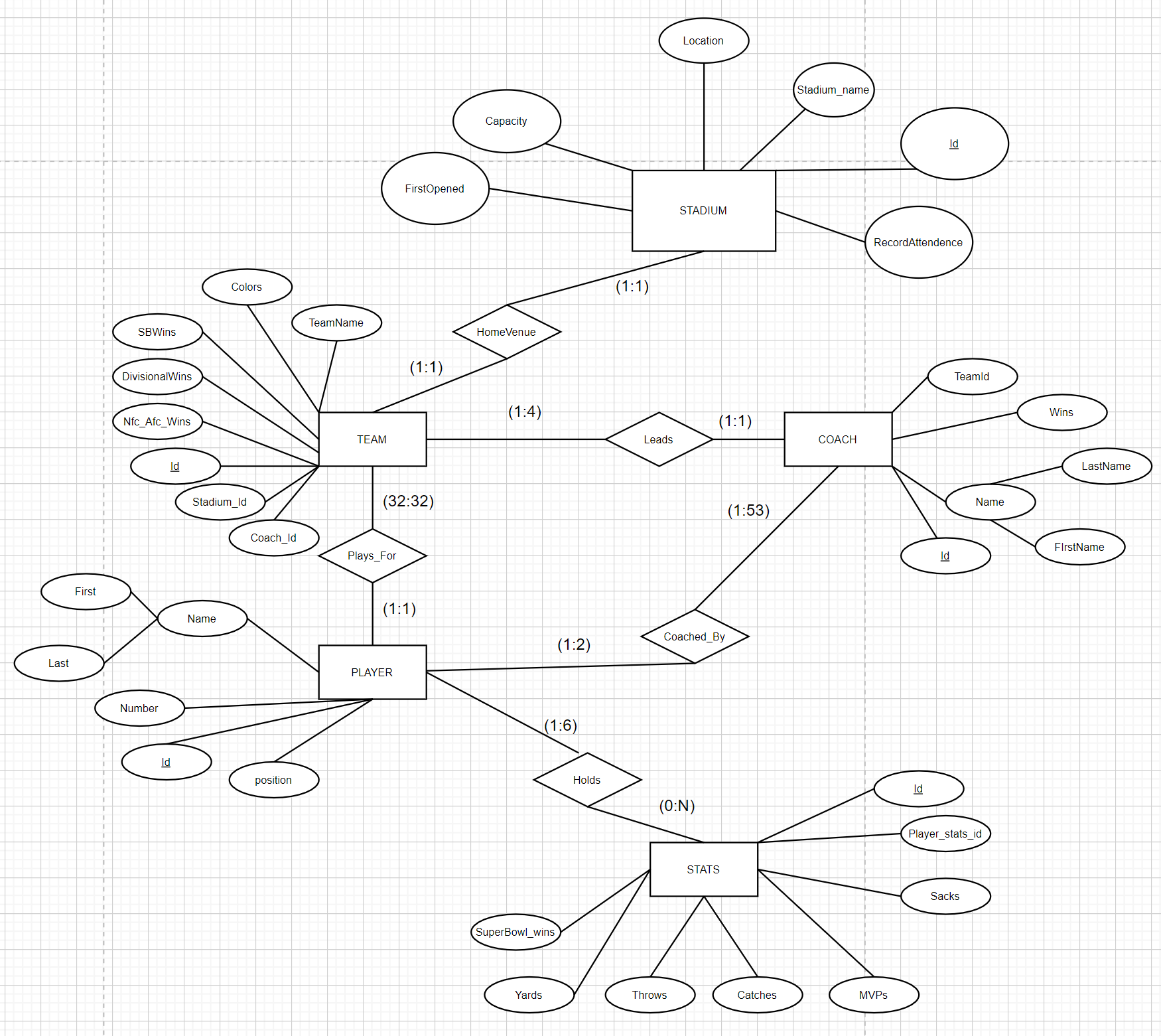
* Report
  + Which includes the documentation explaining the database, including its deficiencies, normalizations, corrections done to the ER diagram and data schema and a future works section.
* Testing Log
  + Which is a spreadsheet or table showing what is expected and how it will be tested.
* Code & Database
  + The DDL that is used to create the database and the constraints and relations.
  + The DML that is used to add data into the database.
  + This also includes the views, tables, triggers and queries.

**ER Diagrams & Schemas**

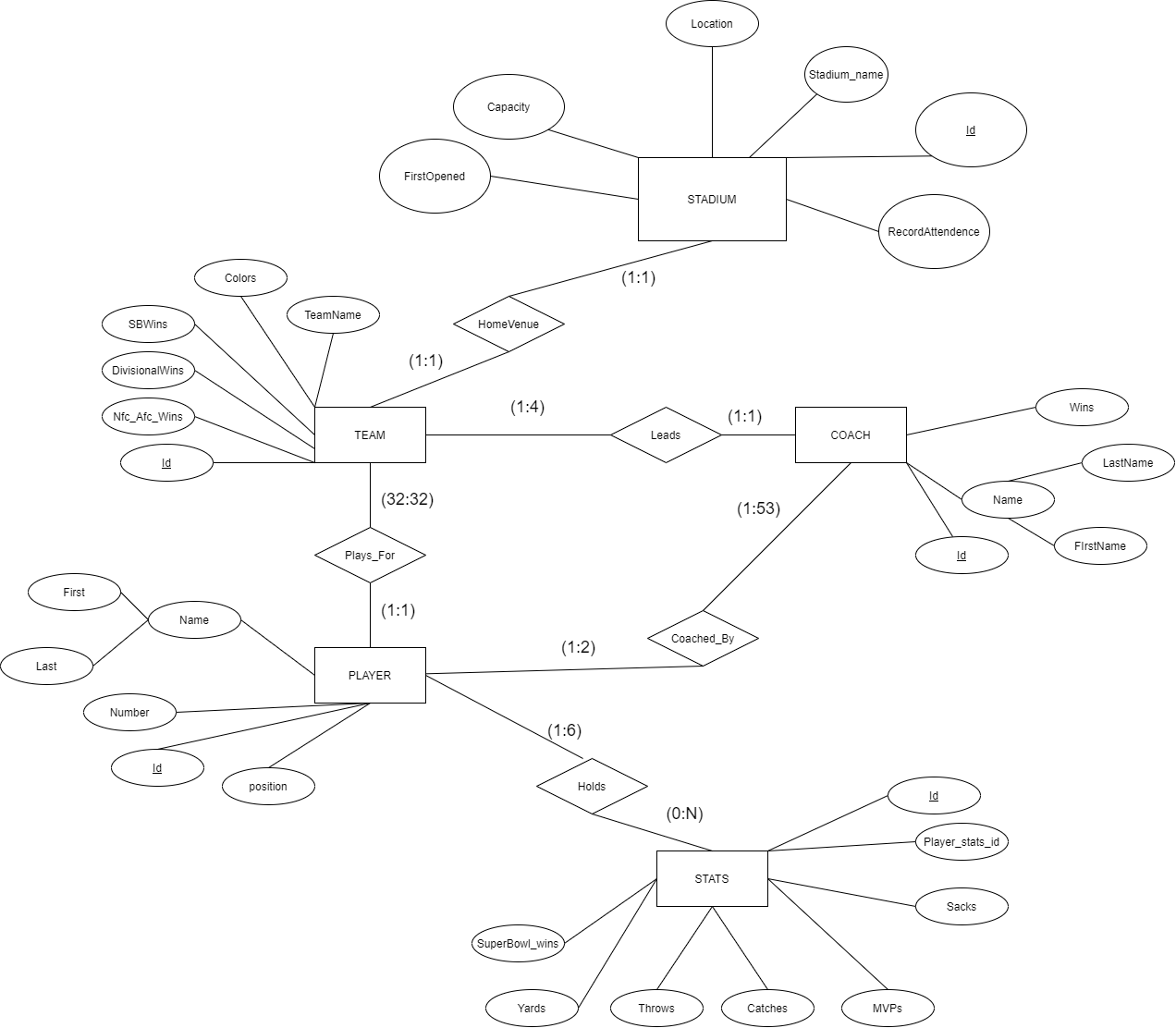
Original ER Diagram



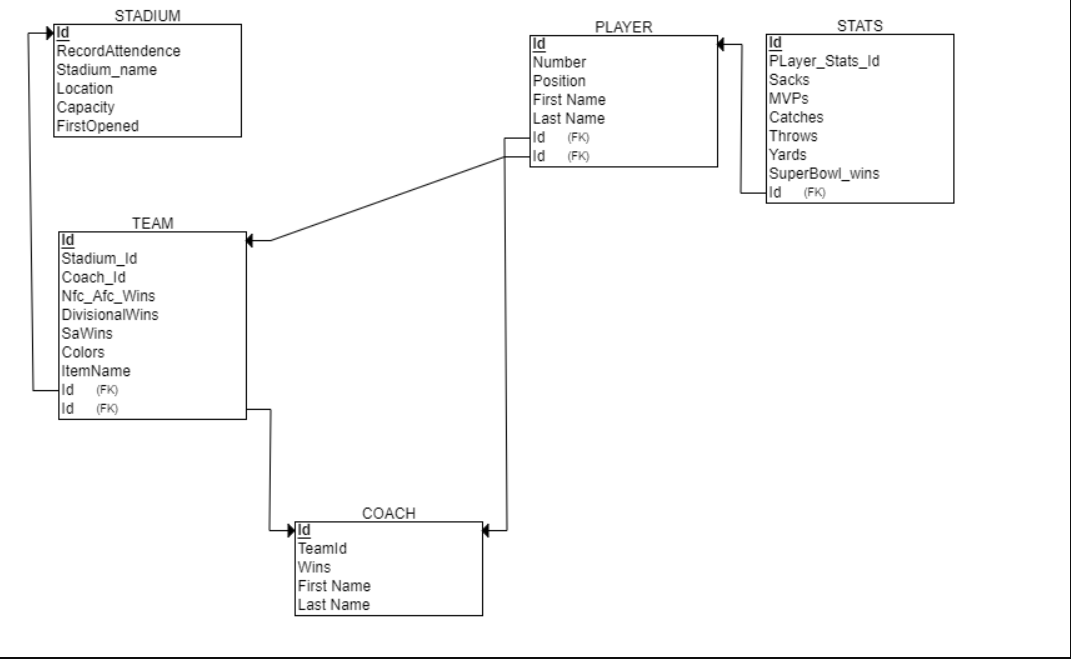
Updated ER Diagram



Normalized ER Diagram



Updated Schema

****

**Normalized Relations**

Before normalization, we had an extra table called NFL\_Records, which was originally used for recording the records made by players. We quickly released that after implementing the database that we could just remove it completely, and instead be able to retrieve that same information by using queries. This resulted in removing redundant data as certain records may have possibly been in multiple tables.. For example Tom Brady’s record for most Super Bowl wins, would have been recorded in both the Stats table and the NFL\_Records table.We also removed some attributes that were used to reference other tables such inside, such as our Team’s coach\_id and team\_id. Our database also meets the requirements to be in second normal form, as non-keys attributes relate only to the keys and are fully functionally dependent on their primary keys.

**Deficiencies**

Here is a list of some deficiencies in our database:

* Lack of visualization of data through an application
* No user roles such as administrator
* Lack of security because of the lack of an application
* Lack of indexes